Response to Office Action (mailing date 10-03-2007)

Application No.: 10/708,757
Applicants: SHANNON ET AL
Examiner: Kishin G. Belani
Art Unit: 2143

For claim 16, we change "domains are derived" to "domains is derived".

For claim 18, we remove the superfluous characters "QuickMarkQuickMark". In passing, please note that we believe these were mistakenly added by the PTO software, as part of the XML markup, when we filed our application.

Regarding claim 1:

Kephart's method differs substantially from ours in several details.

Firstly, consider how a message body is reduced to an invariant form. In Kephart, "the extracted body is transformed into an 'invariant' form by removing all non-alphanumeric characters and replacing all uppercase letters with their lowercase versions (see FIG. 6)." This is the only detailed explanation given in that invention.

In contrast, we refer the examiner to our section on "Canonical Reduction". In common with Kephart, we do replace uppercase characters with lowercase. But we do much more. In one step, we remove HTML comments if these are present. Because a spammer can put arbitrary alphanumeric content within the comments, that are not visible in a browser, but which have the effect of making the message unique. Under Kephart, the HTML comments remain in their "invariant" form of the message.

Likewise, we replace numerical entities that represent a displayable character with that character. For example, if the message has "a" or "a" then both represent the letter 'a', and a browser will display 'a'. But note that the numerical entities are alphanumeric sequences; i.e. they are ascii. Hence, under Kephart, they will be retained in the invariant form of the message. Our invention explains why spammers can use numerical entities to generate apparently unique messages, that are visually the same to a human reading these in a browser.

We remove all HTML tags. Because, as we explained at length in our invention, a spammer can introduce all manners of spurious variation within these tags. Like putting in non-existent attributes, that have random ascii values. Or by varying the order of attributes within a given tag, across different messages made from the same base message. All these are non-perceptible changes. Kephart retains the tags, and hence is vulnerable to the spammer's actions.

Also, we remove any scripting code. This is written in ascii and would remain under Kephart's method. But it lets a spammer write arbitrary source code, unique to each message, and which has no visible effect to the reader.

Our "Canonical Reduction" section describes numerous other steps that Kephart does not do, to reduce non-visible variability.

The examiner remarked that Kephart "disclos[es] a process of a condensed representation of the message body by eliminating message content not perceptible in the normal display mode of the message". Above, we have shown that Kephart does not do this, at least to the extent that we do. Kephart's steps are very limited, and reduce the ability of his method to count instances of spam.

Quine does not use any invariant reduction steps.

Secondly, Kephart's method of generating signatures differs from ours. They find "at step 506, one or more sequences of characters that are highly unlikely to be found in a typical message are identified. The one or more sequences constitute the signature or signatures." Where "unlikely" is defined by an antivirus method referenced by them. They then find a histogram of occurrences of these sequences in the reduced message, using "n-gram statistics".

Our method is completely different. We do not use any estimates of unlikeliness. Essentially, we chop up our canonically reduced form of the message and make hashes of each non-overlapping subset. There is no making of a histogram of sequence occurrences.

Quine does find a signature from a message. But only if the message contains a digital certificate. In this case, Quine finds a signature using standard known methods, in order to verify the certificate. Typically, such methods involve using the entire message body, with no invariant reduction steps.

Regarding claim 2:

Kephart uses the steps in their processing blocks 606-612 to compare a new message's signatures against their database of existing signatures. Our comparison is described in our section 4.1, "Local Comparisons". Our comparison method is simpler. We just compare the hashes made from the new message with those in our database. We do not use Kephart's RegionData and associated RegionLength. These refer to sections of their reduced message, from which checksums are made. We do not use checksums in our method, because these are not as effective as standard hash methods.

In our "BME", there is no equivalent to the RegionData and RegionLength.

Quine does not do any matching of a new message's "signature", as derived from the message body, against a database of such signatures.

Regarding claim 3:

The examiner remarked that Kephart does a transformation of a message body to a "reduced ('invariant') format by eliminating message content not perceptible in the normal display mode of the message". Unfortunately, Kephart's efforts in this regards are very limited. Please see our comments above regarding claim 1. Our removal of non-perceptible content is far more extensive than Kephart.

Quine does not transform the message to a reduced or invariant format. Thus we suggest there is no intersection between claim 3 and Quine.

Regarding claim 4:

Kephart's use of plural hashes is where they are contained in a HashBlock structure. Where each hash has associated with it a count of the number of times the underlying character sequence occurs in the text. Our use of plural hashes has no such associated count.

Quine does not use any hashes. Here, there is a possibility that an implementation of Quine's method, that verifies a message with a certificate, might compute one hash, and then apply public key methods to that hash. But this would be for one hash made from a message, and not plural hashes.

Regarding claim 5:

Kephart in this regard centers around blocks 502 and 504 in his invention. His reduction of non-communicative information is by "removing all non-alphanumeric characters and replacing all uppercase letters with their lowercase versions". But as we explained in our remarks above about claim 1, Kephart's steps are very limited. Scripting code, HTML tags, numerical entities and other constructs remain in the message, under Kephart, while we remove these.

It should be pointed out here that, for example, HTML tags can possess communicative information, that is used by the browser to display the message. Like showing some text strings in bold font, or in color that is different from the default color of most of the text. But in our invention, in the section on Canonical Reduction, we explained at length why HTML tags can be used to hide non-communicative information.

Quine does not remove any non-communicative information from the message body.

Regarding claim 6:

Kephart uses checksums, of at least 32 bits. We use hashes, with one implementation being SHA-1, which uses 160 bits per hash. Mathematically, hashes are generally considered superior to checksums, as a signature of an underlying bit sequence. The reason is suppose we have 2 sequences, that are different. They can be of different lengths, and, if they are the same length, the contents can be different. Doing a checksum of both is far more likely to give the same result, than doing hashes.

Quine appears to do nothing in this regard.

Regarding claim 7:

Kephart's method does not remove email addresses from the body of a message. An email address is written in alphanumeric characters. As such, in Kephart step 504, the only effect is to convert any uppercase characters in email addresses in the message body to lower case. Now consider the next step in Kephart, 506. This finds "unlikely" sequences of characters in the reduced message body. Suppose that an email address in the body is found by Kephart, as one of these unlikely sequences. The address is now considered to be one of the signatures characterising the message. (Or to be part of one of these signatures.)

This is totally opposite what we do. Our method removes email addresses from the message body, as one of the (optional) steps in achieving an invariant form. The reason is that a spammer might insert a fake address, like "mike012335@yahoo.com" into the body, perhaps to make unique instances of the message. Or, the address might even be for an actual email account, which might or might not necessarily be owned by the spammer. For instance, she might put in email addresses of real, non-spammers.

Our remarks here also apply, with trivial modification, when the addresses in the message body are hyperlinks.

Quine's method, insofar as regards addresses, is mostly directly at analysing email addresses in the header. In the body, the only treatment of addresses appears to be a consideration of hyperlinks. Where, in Quine [0019] for example, there is a brief discussion of comparing link text with the associated URL. But this does not constitute a removal of that link from the message, inasmuch as Quine does not really involve reducing a message to an invariant form.

Regarding claim 8:

Kephart is silent on the extraction of addresses from a message. In Kephart's Detailed Description, the second sentence includes "A bus 12 is comprised of a plurality of signal lines for conveying addresses, data and controls between a central processing unit (CPU)". But

nowhere subsequently in Kephart is there any further mention of "addresses" or links or hyperlinks. Beyond the mere mention of addresses that quoted sentence, it does not appear that Kephart specifically treats this.

It is possible, perhaps, that in the final reduced form made by Kephart, the extraction of "unlikely" sequences might include email addresses or hyperlinks. But if so, this appears coincidential. Whereas we explicitly extract addresses as part of the metadata about a message.

Quine [0013] treats email addresses present in the header. It does not treat email addresses or hyperlinks in the message body. With only 1 caveat. Where it compares the hyperlink text with the URL in Quine [0019]. Quine throws an alert if there is some sort of mismatch.

We differ in that we extract hyperlinks, get the domain addresses, and reduce these to "base domains" [e.g. "test.com"]. We do not test the hyperlink text.

We withdraw claim 9.

Regarding claim 10:

Please see our above remarks for claim 8. Kephart does not appear to treat addresses in the body in any explicit fashion. We also differ from Quine regarding the treatment of addresses in the body.

We withdraw claim 11.

Regarding claim 12:

We amend the phrasing of claim 12, to prepend this: "The method of claim 1, with ".

The reason for doing this is given in our remarks above about claim 1. We described in extensive detail why claim 1 differs from Kephart. Thus, by amending claim 12, the difference is now in how a set of characteristics of a message is found; namely by claim 1. Our amendment should now make explicit the difference with Kephart.

Regarding claim 13:

We amend the phrasing of claim 13, to prepend this: "The method of claim 1, with ".

Please see our remarks for claim 12.

Regarding claim 14:

We amend the phrasing of claim 14, to prepend this: "The method of claim 1, with ".

Please see our remarks for claim 12.

Regarding claim 15:

Please see our remarks for claims 3 and 1. As claim 15 is derived from claim 3, we suggest that it has no intersection with Kephart or Quine.

Barchi has a section, "Message-Based Heuristic Filtering". However, there is only one sentence that describes what this filter might be - "Message-based heuristic filtering attempts to identify undesired e-mail by analyzing segments of the received e-mail message such as special content, headers, addressing style, and sender address." This is very vague and open-ended.

In contrast, our invention has section 6 - "Styles", where by "styles" we mean heuristic rules. Our heuristics are explicit, numerous and computable. For example, one heuristic is if the message body is encoded in base 64 or quoted-printable. Another heuristic is if the message only has images (one or more). Another heuristic is if the message has invisible text (where foreground text color = background color). Another heuristic is if the message has numerical entities with leading zeros. In section 6.1, "Message Styles", we described 23 heuristics, from which these examples were drawn.

In section 6.2, "BME Styles", we described a set of 5 heuristics and another set of 11 heuristics.

In section 6.2.1, "Domain Styles", we described 10 heuristics.

In section 6.2.2, "Sender Styles", we described 3 sets of 8 heuristics.

There is a vast difference between Barchi and our invention.

Regarding claim 16:

Kephart does not make a blacklist. In Kephart's "Background of the Invention", there is a mention of blacklists in the section "Domain-based Detection". But this is a description of the art

prior to Kephart, and not about anything in the Kephart invention.

Quine does not make anything like our bulk message envelopes. Nor does Quine construct a blacklist.

Our claim 16 refers to the construction of a blacklist, and not its usage. Huang has a real time blacklist checker module 504. However, Huang [0086] reveals that their blacklist is obtained from a source external to their invention, "The RBL Checking Module 504 can be implemented with products from Mail Abuse Prevention System (MAPS) LLC." Module 504 does not construct a blacklist. Likewise, no other part of Huang does so. Thus, our claim 16 has no intersection with Huang.

We change claim 17 to the following.

17. A method where a blacklist is applied against messages, by finding addresses in links in the message body; if a link has an address in the blacklist, then the message is characterized as "unwanted" or "spam".

We also make the following remarks, which are not part of claim 17, but which might assist the examiner.

Huang uses a blacklist. But it is applied to IP addresses in the header of a message (Huang [0079]), or to the IP address of a mail sending program on another computer, that is connecting to a mail server running Huang's method (Huang [0086]). Huang never applies a blacklist to hyperlinks in the message body.

Quine uses a blacklist in this manner: "the sender address can be compared to a database that includes a list of known spammer addresses" (Quine [0039]). The sender address is from the message header. Quine never applies a blacklist to hyperlinks in the message body.

In Quine [0019], there is a comparison of the hyperlink text with the actual hyperlink URL. There is no usage here of a blacklist. In passing, note also that the efficacy of this paragraph in Quine is very limited. It states in part that "A comparison is made between the first element of the electronic communication and the second element of the electronic communication. An alert condition is established when the first element and the second element do not correspond". But typically the text and the URL will have little in common anyway, for typical links in HTML documents, be these web pages or email bodies written in HTML. Consider a common text "Click here" or "Information" or "Help" with an accompanying URL of "http://anywhere.com". Under Quine's method, an alert would be raised in each case.

Another consideration is the following. In 1998, Schwartz and Garfinkel wrote a book, "Stopping Spam" (ISBN 156592388X, O'Reilly). The book shows that already in 1998, spam was a serious enough problem to warrant a book on antispam methods. The use of blacklists is

described in it. But only against sender and relays in email headers, and against mail servers connecting to a given mail server. However, much spam in 1998 was already written in HTML, with links to spammer domains. Claim 17 could have been usefully applied in 1998, quite separately from the rest of our invention.

One of us (Boudville) asked one of the authors (Schwartz, alansz@uic.edu) about the method. He replied (private communication, 15 September 2007) that "as far as I know, no one was doing url-checking message bodies back when that book came out". We consider Schwartz an expert on antispam, as attested by his co-writing of that book, and by his development and editing of the free and widely used Spam Assassin software.

(The examiner can search the web for the above book and Schwartz and ask him, independently of us.)

It was not until late 2003 and early 2004 that major ISPs started applied blacklists to links in message bodies. There was no technological impediment to them doing this in 1998, had they thought about it then.

Hence we suggest that claim 17 is a priori non-obvious.

Regarding claim 18:

Kephart does not propagate a blacklist to routers, gateways or relays. In our above remarks about claim 16, we pointed out that Kephart does not make a blacklist. Thus, it makes little sense for them to propagate it, and the text of Kephart confirms this.

Quine does not make or propagate a blacklist to routers, gateways or relays.

Huang uses a blacklist from an external source, but does not propagate it to routers, gateways or relays.

Engberg defines a Trusted Party in its paragraphs [0160-0167]. We suggest that its functionalities are not those of a router, gateway or relay, where these latter 3 terms have their common meanings in computer technology. A TP intermediates between a Company and a Client, where the interactions are defined in Engberg. These are specialised interactions. But, a TP does not intermediate generic Internet traffic between 2 parties, who are not acting in the roles of Company and Client (as defined by Engberg).

Thus, we suggest that claim 18 has broader scope than Engberg, in this respect.

Regarding claim 19:

It extends claim 3. Please see our above remarks on claim 3. We suggest those negate any intersection with Kephart and Quine.

Rounthwaite mentions user preferences, but there is no mention of a user being able to have a list of approved or desired senders, whether these senders send bulk messages or not. Please see our remarks below on claim 22 for further elaboration.

We withdraw claims 20 and 21, and replace them with the following claims 24-28. We also make the following remarks, which are not part of the claims, but which might assist the examiner.

The examiner remarked "Kephart, as modified by Quinn et al, further shows and discloses the claimed method, in which these plural hashes may be exchanged by different organizations or users in an anonymous query manner that preserves the privacy of the original messages". Kephart refers to the comparing to signatures (as defined by his method) found from different messages, to see if the messages are essentially similar or same. For this, he uses a "master signature database", which may optionally update "local signature databases". The latter implicitly trust the master signature database. (The locals and master might be run by the same company, for example.)

Our claims below allow for a configuration of independent master signature databases, where these can be expected to lie to each other. Each master database might be run by a different company. Under these conditions, as elaborated in our Application Section 4.2.4, spoofing might occur, where one party falsely claims to have seen a message whose hashes were sent to it from another party. In Kephart's method, there is simply no way to detect this spoofing. The entities that implement Kephart's method are assumed to trust each other, regarding the veracity of the information sent between them.

Put simply, let Alpha and Beta run Kephart's method. These are different companies. Alpha sends hashes {h} to Beta, where these have been derived from a message that Alpha got. Alpha asks if Beta has seen this message, or an equivalently similar message that produces the same hashes. Beta says "yes", even though it has not actually gotten that message. In Kephart's method, there is no way for Alpha to tell that Beta is lying, without Beta actually transmitting an original message to Alpha. Where it can be expected that Beta will refuse to do this, in the name of protecting privacy, whether or not Beta actually has a copy of the message.

It might be asked, why would Beta lie to Alpha? If Kephart's method is used for antispam, it must be expected that spammers might attack it. One mode is where a spammer can join a set of parties using the method, and then proceed to contaminate the results of the method, as these are shared between the parties. For example, if Beta got hashes or signatures for an "original" (i.e. non-spam) message from Alpha, it might reply that it also has seen that message. This could incline Alpha to treat that singleton message as spam. Which acts to discredit the entire method.

A utility of our claims below is that it permits independent parties to query each other's databases

for the existence of messages, without passing the original contents of those messages to another party, and, via subhashes, to check that the other party does indeed possess a copy of that message.

Kephart's method assumes that the parties trust each other, or, equivalently, that they come with some mutually acceptable reputations, where these reputations are provided by mechanisms external to Kephart. In contrast, our method does not require the parties to have these trusted reputations. Indeed, our parties can be anonymous, which lends itself to the construction of peer-to-peer networks for antispam purposes.

24. A method in which various canonical steps are applied to a message, to make an invariant form, thence from which 1 or more hashes and "subhashes" are made; where for each text, in the invariant form, from which a hash is made, another hash is made (the "subhash"), where the subhash is generated from a different starting location in the text.

25. The method of claim 24, where this is performed by different users or ISPs for incoming and outgoing messages, and where the same canonical steps are done by these parties.

26. The method of claim 25, where party A asks party B if B has seen a message with a given set of hashes, where this set is transmitted from A to B; and where, if B replies "yes", B also transmits the corresponding subhashes that it has computed.

27. The method of claim 26, where if the subhashes that A gets from B do not correspond to those that A computed, then it considers that B has not gotten a copy (or a similar copy) of the message that A received.

28. The method of claim 27, where a peer-to-peer network is built, possibly for antispam purposes, where privacy is preserved and where mutual verification is done in an automated fashion; and where parties might be anonymous.

Regarding claim 22:

It extends claim 3. Please see our above remarks on claim 3. We suggest those negate any intersection with Kephart and Quine.

Concerning Rounthwaite - As the examiner has pointed out, Rounthwaite does have a button, "solicited commercial email". This lets the recipient of an email pick the button, to tell the mail server that she wants this email and that she regards it as "commercial", which is essentially equivalent to "bulk" in our invention. Now it can be imagined that were she to do so, the server would then add the sender to a "gray list" for that user.

However, that button is shown to her only when the server picks a given message as being "selected for polling". The polling consists of presenting the message and several buttons,

including the above button, so that users can vote on the message. If a user is sent a message by a bulk sender, and she want this, but it is not polled, then there is no a priori means for her to maintain a gray list, with that sender in it.

Regarding claim 23:

In our invention, we defined an Electronic Communication Modality (ECM). It basically means a type of electronic messaging. We listed several of these - email, Instant Messaging (IM), Short Message Services, peer-to-peer interactions. Plus, in the context of our invention, web pages would constitute another ECM.

The gist of claim 23 is that we could, for example, find Bulk Message Envelopes (and data derived from these, like a blacklist) from a set of email messages. Where the latter are one type of ECM. Then, we could apply that data in another ECM. For example, a blacklist found from the former ECM could be used to block requests from a web browser to web servers located at addresses in the blacklist. Or the email blacklist could be used by an IM server, to block IM coming from addresses in the blacklist.

In Kephart, the use of the Local and Master Signature Databases is just with respect to one ECM, email. The Local and Master Databases distribute data between themselves. But the data are ultimately coming from one ECM, email. And they are used to check new, incoming email, which is the same ECM.